# Learning Task Sequences from Scratch: Applications to the Control of Tools and Toys by a Humanoid Robot

Artur M. Arsenio
MIT CSAIL
Email: arsenio@csail.mit.edu

*Abstract*— **The goal of this work is to build perceptual and motor control systems for a humanoid robot, starting from an infant's early ability for detecting repetitive or abruptly varying world events from human-robot interactions, and walking developmentally towards robust perception and learning.**

**This paper presents strategies for learning task sequences from human-robot interaction cues. Demonstration by human teachers facilitates robot learning to recognize new objects, such as tools or toys, and their functionality. Self-exploration of the world extends the robot's knowledge concerning object properties. Multi-modal percepts are then acquired and recognized by robotic manipulation of toys and tools.**

## I. Introduction

There is a large spectrum of applications for flexible robotic tool handling in industrial environments or for service robots. Children start learning and developing such competencies by playing with toys. Hence, toys are widely used throughout this work as learning aids. The operation of handling tools, such as hammers or swiping brushes, requires not only object location and recognition algorithms, but also algorithms to learn tasks executed with such tools. Therefore, task sequences will be learned on-line from the visual observation of human teachers. Previous approaches for transferring skills from human to robots rely almost exclusively on haptic interfaces [1], [2] for detecting human motion. Environments are often over-simplified to facilitate the perception of the task sequence [3], [4]. Other approaches based on human-robot interactions consist of visually identifying simple guiding actions (such as direction following, or collision), for which the structure and goal of the task are well known [5]. Throughout this paper, task identification and object segmentation/recognition occurs while tasks are being executed by a human teacher without any perceptual over-simplification or environmental setup.

The work presented in this paper was developed on the humanoid robot Cog [6], shown in Figure 1 sawing a piece of wood using neural oscillators to control the rhythmic movements, [7]. According to [7], the robot did not know how to grab the saw or the underlying task sequence. The neural oscillator parameters need to be inserted off-line, using a trial-and-error approach. An automatic approach for selecting the parameters was proposed in [8]. But it remains necessary to recognize the object - a saw, identify it with the corresponding action, and learn the sequence of events and objects that characterize the task of sawing (described in Section II). Furthermore, a general strategy should apply to any object the robot interacts with, and for any task executed, such as hammering or painting. Other research work [9] described robotic tasks such as a robot juggling or playing with a devil stick (hard tasks even for humans). However, the authors assumed off-line specialized knowledge of the task, and simplified the perception problems by engineering the experiments.

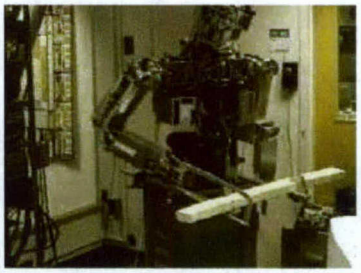

Fig. 1. The humanoid robot Cog has two, six degree-of-freedom (6-dof) arms and a 7-d.o.f head. Each joint is driven by a series elastic actuator [7]. Hence, this compliant arm is designed for human/robot interaction.

This paper describes an embodied approach for learning task models while simultaneously extracting information about objects. Through social interactions of a robot with an instructor (see Figure 2), the latter facilitates robot's perception and learning, in the same way as human teachers facilitate children perception and learning during child development phases. The robot will then be able to further develop its action competencies (as introduced in Section IV), to learn more about objects (Section III), and to act on them using simple actions such as shaking (described in Section V).

## II. Extracting Task Knowledge

The world surrounding us is full of information. A critical capability to an intelligent system is filtering task relevant information. Figure 3 shows events detected from the demonstration of a repetitive task – hammering a nail.

### A. Tasks as Hybrid Markov Chains

Tasks are modelled through a finite Markov Decision Process (MDP), defined by five sets $< S, A, P, R, O >$. Actions correspond to discrete, stochastic state-transitions a $\in$ A={Periodicity, Contact, Release, Assembling, Invariant Set, Stationarity} from an environment's state $s_i \in S$ to the

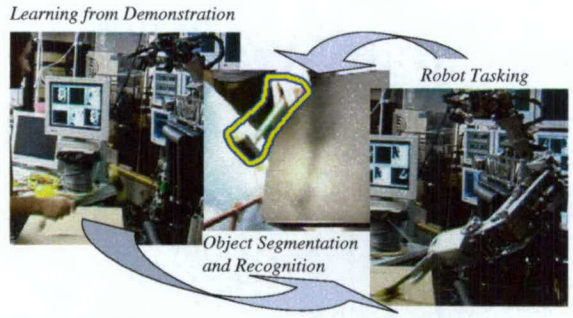


Fig. 2. At initial stages, object properties (e.g. appearance, size or shape) are unknown to the robot. As a human teacher manipulates objects, the robot build object models and learn the teacher actions. Robot learning may then continue autonomously.

next state $s_{i+1}$, with probability $P^a_{s_i s_{i+1}} \in P$, where P is a set of transition probabilities $P^a_{ss'} = P_r\{s_{i+1} = s'|s,t\}$. Task learning consists therefore on determining the states that characterize a task and mapping such states with probabilities of taking each possible action.

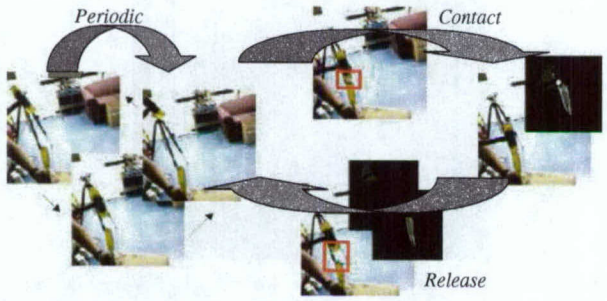


Fig. 3. Hammering task. This task is both characterized by periodic motions, and also by spatial, discrete events. Through observation of the task being executed, the robot learns the sequence of events that compose a task, as well as the objects being acted on.

### B. State Space

States correspond to the continuous dynamics of a system, and are defined as a collection of objects $o \in O$ (O={hammer, nail, actuator,etc}) and a set of relations $r \in R$ between then (R={not assembled, assembled, in contact} - undergoing work is extending this set to account for kinematic constraints between objects, such as revolution or prismatic joints). Information concerning assembling properties among objects is used to build hierarchical object relationships.

### C. Transitions

Sequence of images are analyzed at multiple time/frequency scales for the detection of periodic or discrete events caused by an agent's actions [10]. Transition statistics are obtained by executing a task several times. An action's frequency of occurrence from a given state gives the transition probability.

*1) Perceiving Repetitive Actions:* The detection of periodic transitions created by human teachers, such as hammering a nail, is applied at multiple, logpolar scales. Long spatial intervals (and hence small window sizes) provide more precise spatial, local information, while larger windows increase frequency resolution.

TABLE I

CATEGORIES OF DISCRETE, SPATIAL EVENTS

| *Type of Interaction* | Contact<br>*eg. poking/grabbing an object, assembling it to another object.* | Release<br>*eg. throwing or dropping an object, or disassembling it* |
|---|---|---|
| Moving objects | •overlap of two entities<br>•large a priori velocities | •two moving entities loose contact<br>•large a priori velocities |
| Stationary objects | •abrupt grow of the actuator's motion area<br>•large actuator velocity<br>•abrupt velocity rise for previously stationary object | •large initial velocity of ensemble<br>•large a posteriori velocities for at least one of the entities<br>•motion flow of assembled region separates into two disjoint regions |

A grid of points homogeneously sampled from a moving region in the image is initially tracked over a time interval of approximately 2 seconds (65 frames). The motion trajectory for each point over this interval is determined by Lucas-Kanade pyramidal algorithm. A Short-Time Fourier Transform (STFT) is applied to each point's motion sequence. Periodicity is estimated from a periodogram determined for all signals from the energy of the STFTs over the spectrum of frequencies. These periodograms are processed by a collection of narrow bandwidth band-pass filters. Periodicity is found if, compared to the maximum filter output, all remaining outputs are negligible [11].

*2) Perceiving Discrete Actions:* Time information is lost by transforming a signal to the frequency domain. However, signals generated by most interesting moving objects and actors contain numerous transitory characteristics, which are often the most important part of the signal, and Fourier analysis is not suited to detect them. Therefore, it is imperative to detect events localized over the temporal sequence. In standard video compressing systems, only changes with a significant content over a stationary scene are stored from a sequence of images. From this perspective, good spatial event candidates are moving regions of the image that change velocity abruptly under contact, as shown in Figure 3.

The algorithm to detect such variations works as follows. A motion mask is first derived by subtracting gaussian filtered versions of successive images. A region filling algorithm is applied to separate the mask into regions of disjoint moving non-convex polygons (using a 8-connectivity criterion). Each of these regions is used to mask a contour image computed by a Canny edge detector. The contour points are then tracked using the Lucas-Kanade algorithm. An affine model is built for each moving region from the position and velocity of the tracked points. The motion is then used to test for categories of discrete transitions (see Table I).

### D. Goal Inference

The Markov chain jumps to a final state whenever the environment is stationary, or else whenever an invariant set

is reached. Invariant sets are defined as a sequence of transitions which are repeatedly executed along the Markov chain. Invariant sets are detected by tracking sequences of actions and testing them for loops. All states that belong to an invariant set jump to a final state, as shown in Figure 4.

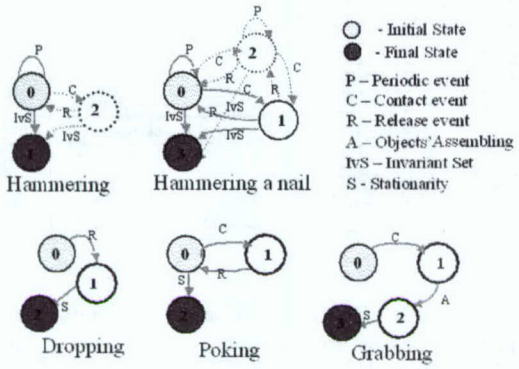


Fig. 4. Hybrid Markov chains for different tasks. The chain for simply waving an hammer contains just two states. But an error occurred for one of the experiments in which an additional state, arising from a contact event with the hammer's own shadow, is created. Since tasks are executed from different locations and light conditions, this error is not statistically relevant. For the task of hammering a nail, contact between the hammer and a nail creates a transition from state 0 to state 1, while a release creates an opposite transition. An additional state is created whenever another actuator is holding the nail. The other graphs correspond to simple, non-oscillatory actions.

## III. Learning about Objects

Both object segmentation and recognition problems are casted under a developmental framework [12]. This strategy permits learning models of objects first from experimental human/robot manipulation, and their a-posteriori identification with or without the agent's actuation.

### A. Extracting The Appearance of Objects and its Boundaries

The set of non-skin moving points tracked over time are sparse, and hence an algorithm is required to group then into a meaningful template of the object, as follows. First, an affine flow-model is applied to the flow data to recruit other points within uncertainty bounds. Clusters of points moving coherently are then covered by a non-convex polygon – the union of a collection of locally convex polygons [11]. This algorithm is much faster than the minimum cut algorithm [13], and outputs segmentations (Figure 5) of similar quality to the active min-cut approach in [14]. Boundaries extraction follows straightforward, by taking edge gradients from a binary segmentation mask. A deformable contour is then attracted to the object's boundary to improve segmentation quality.

### B. Object Recognition

Recognition of objects has to occur over a variety of scene contexts. This led to the development of an object recognition scheme to recognize objects from color, luminance and shape cues, or from combinations of them. The recognition strategy consists of three independent algorithms. The input space for each of these algorithms consists of different features:

Fig. 5. Samples of object segmentations. (left) Top row shows original images, while bottom row shows segmentations (right) Segmentations from a large corpora consisting of tens of thousands of computed segmentations.

*Color:* Input features consist of groups of connected regions with similar color

*Luminance:* Input space consists of groups of connected regions with similar luminance

*Shape:* A Hough transform algorithm is applied to a contour image (which is the output of a Canny edge detector). Line orientation is determined using Sobel masks. Pairs of oriented lines are then used as input features

Geometric hashing [15] is a rather useful technique for high-speed performance. In this method, quasi-invariants are computed from training data in model images, and then stored in hash tables. Recognition consists of accessing and counting the contents of hash buckets. An adaptive Hash table (a hash table with variable-size buckets) was implemented to store affine color, luminance and shape invariants (which are view-independent for small perspective deformations) (see [12] for this algorithm's details and experimental results).

## IV. Control Integration

The multi-scale time-frequency analysis developed in Section II-C offers an elegant solution for integrating oscillatory and non-oscillatory tasks, or mixtures of both. Indeed, tasks can be communicated to the robot with simultaneous frequency and spatial desired contents. The integration of both rhythmic and non-oscillatory control movements is then driven by such information stored in the task description.

### A. Learning Proprioceptive Maps

Controlling a robotic manipulator on the cartesian 3D space (eg. to reach out for objects) requires learning its kinematics – the mapping from joint space to cartesian space – as well as the inverse kinematics mapping.

*1) Visual Maps:* The *Intel Calibration Library* is used to both detect the corners of a calibration object inserted at the robot's end-effector, and to compute the intrinsic and extrinsic camera parameters. Data for computing the camera intrinsic parameters is collected by moving both the arm grip and camera (see Figure 6-1) over a sequence of $n = 20$ images.

A calibrated camera enables the extraction of the extrinsic parameters from the calibration object on the robot's arm grip. Inverting such transformation gives the transformation from the camera's focal center relative to the world referential on the grid $(R_{wh}, P_{wh})$. Hence, by moving the 7-dof robotic head around a stationary arm grip (see Figure 6-2), a map from head joints configurations to the 3D cartesian topological space is

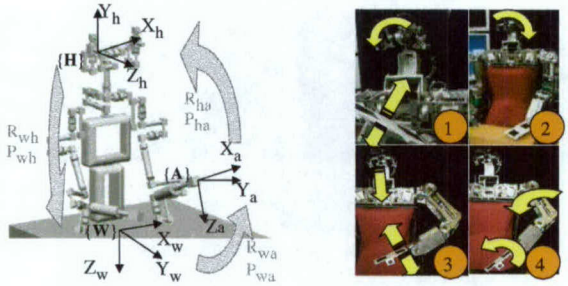


Fig. 6. Learning proprioceptive maps. (left) {W}orld, {A}rm and {H}ead frames, and transformations among these frames (eg. $R_{wa}$ stands for the orientation of frame {A} relative to {W}). (right) Motor behaviors for calibrating the camera (1), learning the head kinematics (2), learning the arm kinematic models (3) and the arm dynamics (4).

TABLE II

MEAN SQUARE ERRORS. 1300 VALIDATION POINTS (50% OF TOTAL).

| Mse | Head Map | Arm Kinematics |
|---|---|---|
| X (cm²) | 177 | 224 |
| Y (cm²) | 75 | 173 |
| Z (cm²) | 74 | 381 |
| $L_2$ norm | 326 | 778 |
| Pitch | 5.81 | 17.15 |
| Yaw | 5.75 | 62.78 |
| roll | 2.46 | 16.35 |
| $L_2$ norm | 14.02 | 96.29 |

| mse | Inverse Arm Kinematics |
|---|---|
| $J_1$ | 11.6 |
| $J_2$ | 4.5 |
| $J_3$ | 17.41 |
| $J_4$ | 10.17 |
| $J_5$ | 9.32 |
| $J_6$ | 2.63 |
| $L_2$ norm | 61.64 |

| mse | Retinal plane (240 × 320) |
|---|---|
| x | 117.73 |
| y | 215.78 |
| $L_2$ norm | 215.88 |

acquired. The calibration object on the arm grip is kept on the camera's field of view by visually tracking it's centroid. A PD controller controls the eyes, while inertial data is used to stabilize the robotic head and for VOR compensation.

The Forward Kinematics mapping the head joints' configuration (given by 6 parameters – just one eye is required) to the cartesian position and orientation of the camera's focal center relative to a world referential, is estimated using locally affine models. Learning of these models was carried out by applying a modified version of memory-based Locally Weighted Regression [9].

*2) Sensory-Motor Maps:* A calibrated camera and a head kinematic map enable the estimation of both the arm forward and inverse kinematics. This is accomplished by moving the robotic arm over the joint configuration space, while visually tracking it's grip (see Figure 6-3). The cartesian location (6 parameters) is determined from two transformations: one from the grip to the camera's focal center $(R_{ha}, P_{ha})$, given by the camera's extrinsic parameters, and the other from the focal center to the world referential, given by the head forward kinematics $(R_{wh}, P_{wh})$. This data, together with the 6 joint angle measurements, generates training data to build locally affine models using memory-based Locally Weighted Regression [9] (off-line, batch minimum least squares over locally linear models). Statistical results are shown on Table II.

*a) Locating the Arm on the Retinal Plane:* Detection of the robotic arm's end-effector on the retinal plane follows from both head and arm forward kinematics. From the head/arm joint configurations, the maps predict $(R_{wh}, P_{wh})$ and $(R_{wa}, P_{wa})$, respectively, from which $(R_{ha}, P_{ha})$ follows. The grip location on image coordinates is then just the perspective projection of $P_{ha}$. Table II presents statistical data from building all these sensory-motor mappings.

### B. Controllers

*1) Neural Oscillators:* Tools and toys are often used in a manner that is composed of some repeated motion – consider hammers, drums, saws, brushes, files, rattles, bells, etc. Section II-C.1 introduced a framework to detect simple repeated visual events at frequencies relevant for robot-human interaction [10]. But it is equally necessary to not only perceive the repetitive motions that facilitate the robust perception of these objects, but also to control such rhythmic motions for robot tasking.

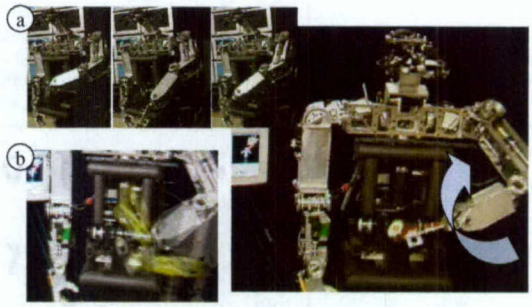


Fig. 7. Cog playing musical instruments – a rattle (a) and a tamborine (b) – using neural oscillators with proprioceptive feedback.

Oscillatory motions are therefore controlled using Matsuoka neural oscillators, consisting of two mutually inhibiting neurons. These neural oscillators are able to entrain the frequency of input signals or resonance modes of dynamical systems minimizing the cost of the actuator energy. We proposed a mathematical analysis for multiple nonlinear oscillators connected to a (non)linear multi-variable dynamic system, by using multiple input describing functions [8]. As a result, the framework developed provides estimates for the frequency, amplitudes and/or parameters of oscillation of the controlled system, as well as an error bound on the estimates, using algebraic equations [16]. Such framework is used to design the robot's oscillatory motions, as shown in Figure 7 for the playing musical instruments: a tamborine and a rattle.

*2) Sliding Modes Controller:* The dynamics of an arm manipulator is strongly nonlinear, and its nonlinear dynamics poses challenging control problems. Especially for mechanisms with small gear transmission ratios or low-reduction cable-driven systems or direct-drive connections, nonlinear dynamic effects may not be neglected. The state space of Cog's $n = 6$-linked articulated manipulator is described by the $n$ dimensional vectors $q$ and $\dot{q}$ of joint angles and velocities,

respectively. Its actuator inputs $\tau$ consist of a $n$ dimensional vector of torques applied at the joints. The nonlinear dynamics can be written as the system [17]:

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \quad (1)$$

where $H(q)$ is the manipulator inertia matrix (which is symmetric positive definite), $C(q,\dot{q})\dot{q}$ is the vector of centripetal and Coriolis torques, and $g(q)$ is the vector of gravitational torques. The feedback control problem for such system is to determine the actuator inputs required to perform desired tasks from the measurements of the system state $(\dot{q}, q)$ of joint velocities and angles, in the presence of uncertainty.

As described in detail by [18], [17], control theories that are most often applied to such systems are Sliding Modes Controllers, PD and Computed Torque Controllers. The latter two are often used to reach desired positions. Sliding modes becomes rather useful to follow desired trajectories specified by $(q_d, \dot{q}_d, \ddot{q}_d)$, the position, velocity and acceleration for each manipulator joint, under model uncertainty.

A sliding Modes control law [18] was implemented, given by equations 2 and 3, where $s$ is a weighted sum of position $(\tilde{q} = q - q_d)$ and velocity errors.

$$\begin{aligned}\tau &= \hat{H}(q)\ddot{q}_r + \hat{C}(q,\dot{q})\dot{q}_r + \hat{g}(q) - K(q_d)sat(\Phi^{-1}s) \\ &= \hat{\tau} - K(q_d)sat(\Phi^{-1}s)\end{aligned} \quad (2)$$

$$s = \left(\frac{d}{dt} + \Lambda\right)^{m=2} \tilde{q} = \dot{\tilde{q}} - \Lambda\tilde{q} \quad (3)$$

The non-linear dynamics is learned adaptively on-line. These non-parametric locally affine models are used to predict the feedforward term $\hat{\tau}$. The reference velocity is given by $\dot{q}_r = \dot{q} - \Lambda\tilde{q}$, and $\Lambda$ is a symmetric positive definite matrix (assumption can be relaxed to -$\Lambda$ Hurwitz). The matrix $\Phi$ defines the boundary layer thickness,

$$\dot{\Phi} + \lambda\Phi = K(q_d) \quad (4)$$

leading to tracking to within a guaranteed precision $\varepsilon = \Phi/\lambda^{m=2}$. Three factors impose upper bounds on $\lambda \approx \lambda_R \approx \lambda_s \approx \lambda_D$: structural resonant nodes ($\lambda \leq \lambda_R$); time delays ($\lambda \leq \lambda_D = 1/T_D$), where $T_D$ is the largest unmodeled time delay (which was set to one sampling interval - 5 ms); and sampling rate ($\lambda \leq \lambda_s = 1/5\nu_{sampling} = 40Hz$).

### C. Non-parametric Learning of the Manipulator Dynamics

Standard controllers, such as the PID controller typically used in industry, do not require a dynamic model of the system being controlled. But performance is lost at high frequencies and instability might occur for very compliant systems. However, the dynamic model of the manipulator is often unknown or else known with a large uncertainty. The non-parametric learning of the arm dynamics was therefore implemented using an on-line, iterative version of Receptive Field Weighted Regression [9], using as input space $(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$ (since $\ddot{q}$ is unknown), and output space $\tau$. The algorithm compensates for the unknown nonlinear terms (specially gravity terms).

## V. Robot Tasking

Humans are pretty good in understanding visual properties of objects, even without acting on them. However, the competencies required for such perceptual capability are learned developmentally by linking action and perception. Actions are rather useful for an embodied actor, through the use of its own body, to generate autonomously cross-modal percepts (e.g., visual and auditory) for automatic object categorization. This is in contrast with non-embodied techniques such as standard supervised learning requiring manual segmentation of off-line data, imposing thus constraints on an agent's ability to learn.

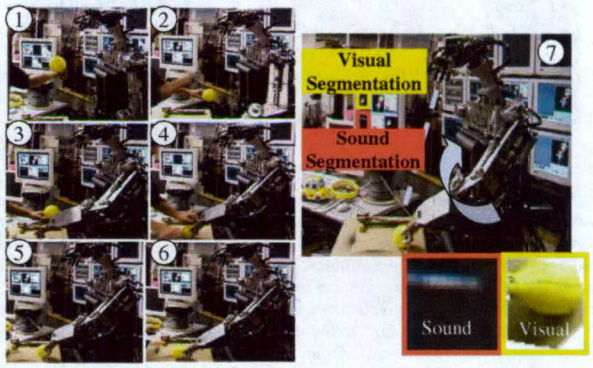


Fig. 8. Reaching to and shaking a child's toy (a Castonete). (1-2) A human actor attracts Cog's attention to the toy, by creating a salient stimulus to its attentional system. (3-4) The robot reaches to the object – feedback control applying a sliding mode controller. (5-6) Cog shakes the toy. Feedback proprioceptive signals are sent into a neural oscillator, which entrains the natural frequency of the dynamic system to which its coupled (the robot's arm), producing rhythmic sounds. (7) Visual and auditory segmentations by Cog's perceptual system. It shows two segmented images - one for a low resolution sound spectrogram over one period of oscillation, and the other for the toy's visual template extracted from the periodic movements of the toy.

Figure 8 shows the humanoid robot Cog executing a task requiring the integration of both reaching movements and rhythmic motions. The task consists of having the robot playing rhythmic sounds with a Castonete by first reaching its grip to the Castonete and therefore shaking it.Such actions enable Cog's perceptual system not only to extract visual descriptions of the Castonete, but also the acoustic pattern that it produces (auditory segmentation and recognition described elsewhere [19]). Percepts from these two different modalities are linked by correlating amodal features – timing and synchrony – through cross-modal processing [19].

Similarly to object segmentations from human demonstration, both the robot's grip and the poked objects are segmented from the robot's actions. Indeed, some segmentation results (e.g., the robot's grip or the lego's brick) shown in Figure 5 were obtained by having the robot poking at objects.

### A. Human-Robot Skill Transfer

We now show how learning a very simple model (just a machine with one internal state) for the task of hammering on a table enables the robot to generate autonomously informative percepts by itself. Consider again Figure 2. These images correspond to real experiments. We have shown before how

object recognition and robot experimental manipulation evolve developmentally from human demonstration. By transferring the manipulation skill from human to robot, the latter can generate equally training data to the object recognition algorithm, as demonstrated by the experiment in Figure 9. This figure shows that by having the robot hammering on a table, the perceptual system extracts visual templates of the object which is thereafter recognized as the same object previously segmented from human demonstration.

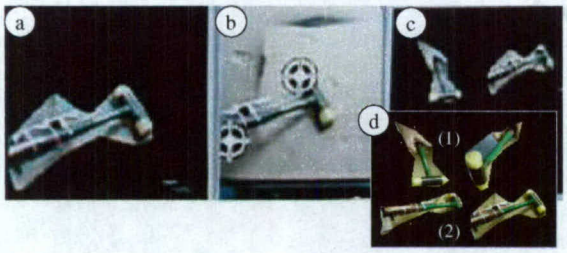


Fig. 9. Human-robot skill transfer, from an on-line experiment. a) Hammer visual segmentation by having the robot hammering on a table. (b) Tracking multiple objects – the robot grip and the hammer – based on the Lucas-Kanade pyramidal tracker algorithm. (c) It shows two segmentations. One first obtained from human demonstration (on the left). The second (on the right), was segmented from robot actuation, and it was recognized as belonging to the same category as the first (otherwise it would not appear on the same window during the experiment). d) Several segmentations obtained by human demonstration and by the robot's experimental manipulation.

### B. Human-Robot Cooperation

Input from one perceptual modality can also be useful to extract percepts from another perceptual modality. This is corroborated by an experiment (see Figure 10) consisting of feeding the energy of the acoustic signal into the feedback loop of the neural oscillator, instead of proprioceptive signals. Therefore, the goal is to have the robot to play drums using sound feedback. The task rhythmic is imposed by a human actor, which cooperates with the robot for drumming with sticks. Since it is difficult for the neural oscillator to engage initially in a rhythmic pattern without a coherent source of repetitive sound, the human guides the process by providing such information. While executing the task, the robot is then able to learn the visual appearance of the drumming stick (shown in Figure 10), together with the sound it produces.

## VI. Conclusion

In this paper we introduced both humans and the robot into the learning loop to facilitate robot perception. The experiments carried out underline the importance of learning to interpret actions. There is a lot to be gained when introducing a human teacher in the loop that guides the robot's learning process. This knowledge is grounded as the robot acts by itself.

## Acknowledgment

Project funded by DARPA as part of the "Natural Tasking of Robots Based on Human Interaction Cues" under contract number DABT 63-00-C-10102, and by the Nippon Telegraph and Telephone Corporation as part of the NTT/MIT Collaboration Agreement. Author supported by Portuguese grant PRAXIS XXI BD/15851/98.

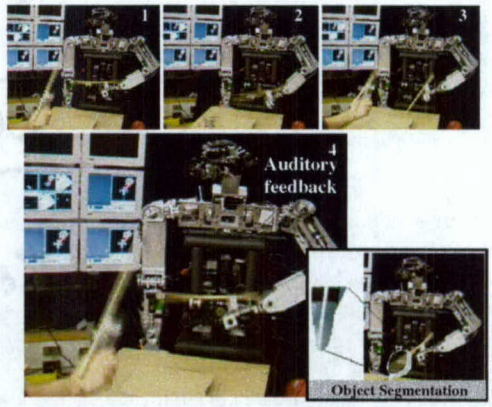


Fig. 10. Human-Robot cooperation. Playing with a stick for drumming, entraining rhythms provided by a human actor, who drums together with the robot. The neural oscillator receives as feedback signal the acoustic energy. The robot is then able to extract a visual segmentation of the stick.

## References

[1] R. Dillmann, O. Rogalla, M. Ehrenmann, R. ollner, and M. Bordegoni, "Learning robot behaviour and skills based on human demonstration and advice: The machine learning paradigm," in *9th International Symposium of Robotics Research*, 1999.

[2] J. Aleotti, S. Caselli, and M. Reggiani, "Toward programming of assembly tasks by demonstration in virtual environments," in *IEEE Workshop on Human-Robot Interactive Communication*, 2003.

[3] K. Ikeuchi and T. Suehiro, "Towards an assembly plan from observation, part i: Task recognition with polyhedral objects," *IEEE Transactions on Robotics and Automation*, vol. 3, no. 10, 1994.

[4] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *Trans. on Robotics and Automation*, vol. 6, no. 10, 1994.

[5] M. Nicolescu and M. Mataric, "Learning and interacting in human-robot domains," *IEEE Transactions on Systems, Man, Cybernetics, special issue on Socially Intelligent Agents - The Human in the Loop, K. Dautenhahn, ed.*, pp. 419–430, 2001.

[6] R. A. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M. M. Williamson, "The Cog project: Building a humanoid robot," in *Computation for Metaphors, Analogy and Agents*, ser. Springer Lecture Notes in Artificial Intelligence, C. L. Nehaniv, Ed. Springer-Verlag, 1999.

[7] M. Williamson, "Robot arm control exploiting natural dynamics," Ph.D. dissertation, MIT, Cambridge, Massachusetts, USA, 1999.

[8] A. Arsenio, "Neural oscillator networks for rhythmic control of animats," in *From Animals to Animats 6*. MIT-Press, 2000.

[9] S. Schaal and C. Atkeson, "Constructive incremental learning from only local information," *Neural Computation*, vol. 8, no. 10, 1998.

[10] A. Arsenio, "Embodied vision - perceiving objects from actions," *IEEE Int. Workshop on Human-Robot Interactive Communication*, 2003.

[11] ——, "An embodied approach to perceptual grouping," in *IEEE CVPR Workshop on Perceptual Organization in Computer Vision*, 2003.

[12] ——, "Teaching a humanoid robot from books," in *International Symposium on Robotics*, Paris, France, 2004.

[13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no. 22, 2000.

[14] P. Fitzpatrick, "From first contact to close encounters: A developmentally deep perceptual system for a humanoid robot," PhD thesis, MIT, Department Elect. Eng. and Computer Science, Cambridge, MA, 2003.

[15] H. Wolfson and I. Rigoutsos, "Geometric hashing: an overview," *IEEE Computational Science and Engineering*, vol. 4, pp. 10–21, 1997.

[16] A. Arsenio, "On stability and error bounds of describing functions for oscillatory control of movements," in *IEEE International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, 2000.

[17] L. Murray and S. Sastry, *Robotic Manipulation*. CRC Press, 1994.

[18] J. Slotine and L. Weiping, *Applied Nonlinear Control*, N. Englewood Cliffs, Ed. Prentice-Hall, 1991.

[19] P. Fitzpatrick and A. Arsenio, *Feel the beat: using cross-modal rhythm to integrate robot perception*. Workshop on Epigenetic Robotics, 2004.